Application and Analysis of the Use of Outcomes for a Higher Education Course Unit

Anthony PRIOR
David GLANCE
School of Computer Science and Software Engineering, The University of Western Australia, 35 Stirling Highway, Crawley, Western Australia, 6009. tone@csse.uwa.edu.au, david@csse.uwa.edu.au, http://www.csse.uwa.edu.au/~tone, http://www.csse.uwa.edu.au/~david

KEYWORDS: *outcomes, validation, HTML-template, reverse engineering.*

ABSTRACT: *The specification of outcomes for a higher education course unit is perceived as bringing many benefits to both students and educators. Traditionally however, the benefits have been seen principally from the perspective of informing students of the learning outcomes and their progress in achieving these outcomes.*

Another major benefit of outcomes is their use as a set of rigorously defined requirements for the unit. The definition of these requirements enables the potential use of outcomes for: [1] the specification of a curriculum that take into consideration subject coverage across units and also coverage throughout a degree programme. [2] Validation of unit content in terms of correspondence with the curriculum. [3] Validation of coverage of unit content by assessment. [4] Specification of outcomes for advanced students.

Outcomes were written for a 1st year introductory unit of Software Engineering taught by the School of Computer Science and Software Engineering at the University of Western Australia. The format of these outcomes was designed with consideration of the above-mentioned uses. A standard HTML based template for each outcome topic was designed which included links from each outcome to the assessment of that outcome. The implementation of this was published as part of the unit homepage.

This paper discusses the process by which these outcomes were written, motivations for doing to and also knowledge gained as a result. Such knowledge includes becoming aware of deficiencies in both the unit content in terms of its coverage of the curriculum and the coverage of the content by the assessment. This subsequently allows for the reverse engineering of the structure of the unit in order to fill any gaps in the learning objectives and/or assessment.

1 INTRODUCTION

Higher education unit and course structure has traditionally been built upon loosely defined goals viewed principally from the educator's and not the student's perspective. It has been recognized that this approach is inadequate and that educational requirements need to be more rigorously defined and stated from the student's perspective [1]. Outcome-Based Education (OBE) provides the means of achieving this, clearly focusing and organizing everything in an education system around what outcomes are expected for all students to have achieved by the end of their learning experience [2].

Once learning outcomes are specified, several other capabilities become possible above the prime goal of providing the student with a clear idea of what they will learn and what level of understanding and ability they are expected to achieve. Firstly, there is a metric by which the overall quality of the education an institution is providing can be measured. This feature was one of the initial drivers for the adoption of OBE in Australia [3]. The notion of quality can be extended to assess individual units as well in terms of coverage of the outcomes by the unit material and assessment of that material. Secondly, the specification of outcomes for all units on a degree course provides a mechanism by which the overall outcomes of a particular degree course can be coordinated and validated [4].

The introduction of outcomes in higher education poses a different set of problems than their use in secondary education. It is recognized that tracking student outcomes is a daunting challenge even for small numbers of students. Given that there may be 300-600 students in any particular class taking a higher education unit, with dozens of individual outcomes, the process of using outcomes as the sole mechanism of tracking student progress becomes logistically impossible. In addition, students require

absolute marks from units for a variety of reasons and so associating absolute marks with achievement levels of outcomes becomes another problem [5].

We undertook to write outcomes for a first year introductory Software Engineering unit given by the School of Computer Science and Software Engineering at the University of Western Australia. The process of producing outcomes was based on deciding on the specific form of the outcome statements (a HTML tabular template). Once this was done, we carried out an initial reverse engineering of the outcomes from the existing unit content. This enabled us to assess the quality of the unit content not only from what was expected from the unit, but also how this unit fitted into the overall teaching of Software Engineering in the degree program. Once the outcomes and content were decided on, they were related to the assessment to verify that the assessment was able to measure levels of achievements of all of the outcomes. The results from this exercise were fed back into the design of the content and assessment of the unit.

As an additional innovation, outcomes were specified for advanced students. With large first year units, there is a wide mix of abilities of the students and some find the base level outcomes easily achievable. By specifying advanced outcomes, better students can optionally focus on these.

2 OUTCOME STRUCTURE

The structure of the outcomes written for the software engineering outcomes was in the form of an HTML template (demonstrated in Figure 1), having four sections for each outcome:

2.1 Unique numbering system

Each outcome had a unique number of the form x.y.z. This enabled readers to associate an outcome with a particular topic taught in the unit (the x component), along with the particular area of the outcome within the topic (the y component) and finally the specific outcome (the z component). Such a numbering system allows for straightforward referencing as each topic is related to the lecture where a particular outcome was taught. Also it is possible that in the future should lecture notes becomes HTML based, they can have hyperlinks to the exact outcome(s) for concepts being taught.

2.2 The written outcomes

Traditional outcomes are written with four criteria [6, 7]:

1. Conditions given under which the student will perform the outcome.
2. Actions the student will perform e.g. recognize, describe, and demonstrate.
3. The particular task the student will be able to do.
4. Manor of assessment of this particular outcome.

We decided to omit the first criteria as many of the outcomes did not befit any specific condition but rather needed to be performed under a variety of situations. Such situations were too numerous or abstract to be incorporated into any specific outcome.

Essentially this section is contrived of the second and third item of the above list. The criteria for this section were taken mainly from the lecture notes that were to be used for the course.

The manor of assessment was left for the next section.

2.3 Assessment of the outcome

This section details where the particular outcome is assessed. If the outcome is assessed in a particular laboratory exercise, a web link to the lab (labs descriptions are also written in HTML and published on the web) is provided for ease of reference. If the assessment included the exam and/or project, this was also stated in this section. Those outcomes that were not assessed according to the current course structure were intentionally left blank at the point the outcomes were written so that it could be seen which areas were not being assessed.

2.4 Level of achievement

Due to the nature of the unit and the diversity of students taking the unit, we chose to implement a specification of achievement for advanced students. At this stage the scheme is split into two levels:

- Level 1 - students are expected to have achieved this outcome should they wish or be capable of receiving a passing grade for the unit.

- Level 2 - students are expected to have achieved this outcome should they wish or be capable of receiving a distinction or higher distinction grade for the unit.

5.4 Methods

Number	Outcome	Assessment	Level (standard=1, advanced=2)
5.4.1	Recognise that methods of a class are described with a signature (name and parameters) and also that two methods cannot have the same signature.		1
5.4.2	Break down the elements of a method signature and describe the function of each	Examination	1
5.4.3	Explain when it is appropriate to use dot notation and when not to when calling another method from within a method.	Examination	2
5.4.4	Demonstrate how to correctly pass parameters to a method (number, position and type).	Lab 1 Exercise 4 Lab 2 Exercise 2 Lab 3 Exercise 1	1
5.4.5	Describe the difference between parameter passing by value and by passing by reference. Discuss how Java approaches this issue.	Lab 2 Exercise 2	2
5.4.6	Describe, with an example, what is meant by returning from a method and what types can be returned.	Lab 1 Exercise 4-5 Lab 2 Exercise 1-2 Lab 3 Exercise 1 Lab 3 Exercise 4-5	1
5.4.7	Describe what a method with the return type **void** will return.		1
5.4.8	Describe method/local variables – where they are used, when they are initialised and how long they last for.	Lab 2 Exercise 4	2
5.4.9	Describe the concept of method overloading and why it may be useful to use this Java function.	Examination	2
5.4.10	Give an example of a class, in particular demonstrate: • Instance/Global • A constructor method • An accessor method • A mutator method	Project	1

Figure1 – Shown is a typical category of outcomes placed written out using the template we developed, in this case Java methods (there is a Java component within the Software Engineering unit). A hyperlink is used when certain outcomes can be linked to the lab exercise where they are assessed. Where certain outcomes have are not assessed under the current course structure, the assessment field is intentionally left blank (figure taken from [8]).

3 METHOD FOR WRITING THE OUTCOMES

Once a suitable outcome structure was devised, we began the task of writing outcomes from the unit. This was done by scrutinizing the lecture notes that are available to each student for each lecture given. The notes for a particular lecture would constitute a unique "x" section of the numbering system. Each of the categories within a set of notes would then constitute a unique "y" section of the numbering system (the table of Figure 1 is one such category). After organizing the categories, the specific outcomes were written (each having a unique "z" number).

In writing the specific outcomes, we attempted to write a selection of standard level 1 outcomes as well as an additional 1 or 2 level 2 outcomes. The decision of which level each outcome should be rated was left at the discretion of the author of a particular outcome. It was judged by our opinion of whether every first year student should be capable of performing the outcome and also whether they would be required to have achieved it in order to undertake succeeding units. Although most outcomes were written from the material of the lecture notes, categories with no level 2 outcomes sometimes had additional outcomes written if we could think of more challenging outcomes relating to that category.

4 RESULTS AND FINDINGS

After performing the first iteration of writing outcomes, we made a number of interesting findings. The most prominent was that there lies a strong degree of difficulty in successfully assessing all outcomes, particularly the more abstract type of outcomes such as outcome 12.1.2, which states, "Appreciate the need for group work given the limitations of single person workgroups" [9]. The sheer number of the more exact outcomes makes assessing all outcomes cumbersome if not impossible. Further still, even if it were possible, this would only be to say that there was an assessment and would not

provide a means whereby the student's level of attainment for any particular outcome could be derived. Many of these type of outcomes could potentially be asked in examination however only so many could be covered in such a short time. This raises questions concerning whether to tailor the outcomes more closely to what is assessed or to alter course structure to encompass the assessment of a wider range of outcomes.

Another finding relevant to the assessment of outcomes was that the level 2 outcomes were for the most part not assessed in the exams of previous years and only assessed to a small degree in the laboratories. If the outcomes are to be used as a tool to directly link student marks to course content and in particular to determine the abilities of different students, assessment content will need to be changed in terms of assessing more of the advanced outcomes.

5 ADVANTAGES OF OBE FOR DEFINING COURSE REQUIREMENTS AND STRUCTURE

The benefit to the teachers and students of OBE has been a well-researched topic [2, 3, 10]. In addition to these, the implementation of outcomes when designing course structure can be a useful tool for administration to assess the validity of course content.

5.1 Specification of the curriculum

Well-defined outcomes can be used when specifying a course curriculum. There are two aspects we consider: that of cross coverage of content across concurrently running units (i.e. units a student is studying at one time) and that of coverage throughout a degree programme (as a student progresses from one semester to the next).

Through examining outcomes of units students are studying at one particular time (or at least those available to the student at that point in their degree), it can be determined whether particular outcomes are being are being (perhaps redundantly) re-taught in different units. The opposite of this is also of concern: desired course outcomes for a particular level being omitted (perhaps through miscommunication between unit organizers).

With consideration of examining content coverage throughout a degree programme, well-defined outcomes can be used to ensure that students will have acquired the necessary knowledge and skills to participate in a higher level of a course as they progress from one level to the next.

It is entirely possible that the outcomes, once defined to a detailed and sufficient degree, can be used as the statement of the curriculum itself. Upon determining the success of using the outcomes as a tool for providing guidelines of expected achievement from students, we may choose to implement the outcomes as the main (perhaps even sole) specification of the curriculum.

5.2 Validation of unit contents in correspondence to the curriculum

Another benefit of implementing outcomes is that it enables administration and teachers to ensure that the content of a particular unit is sufficient to satisfy the learning objectives stated in the curriculum.

The reverse engineering process we undertook to write the outcomes enables us to note deficiencies of the content of the software engineering unit. By comparing the reverse engineered outcomes against a set provided by the curriculum, the course content can be adjusted appropriately to both incorporate missing content and omit any unnecessary content.

5.3 Validation of unit content by assessment

The format of the outcomes written was such that each outcome had a link to the point in the course where that outcome was assessed (if at all). This was usually in the form of a link to the specific laboratory exercises where the various outcomes were addressed. This provided a means to determine which outcomes were being adequately assessed and which were not.

This processes for not always straightforward as the more implicit outcomes were incongruous with such a format. It is the duty of the unit coordinator to ensure that the assessment of these types of outcomes is addressed at through other means such as large projects and examinations.

The format we derived will prove useful for the validation of the assessment criteria, particularly in ensuring that material covered in lectures is assessed adequately through laboratories, project and exams.

6 FUTURE IMPROVEMENTS

As this was the seminal effort to transform the make up of the Software Engineering unit to an outcomes based system, there will be numerous reiterations of the defining of the outcomes.

For the next iteration, we will liaison with the unit coordinators of succeeding units to determine, from the outcomes we have written, which are relevant and more importantly what aspects of programming/software engineering have not been adequately covered in the course.

One of the motivations for writing the outcomes was to use them for determining absolute marks for students. At this stage we have not attempted to do this however this would be something that will be addressed in the next iteration. This would perhaps be through linking marks achieved directly with the attestation of the ability to perform various outcomes.

We have seen that to attempt to link every outcome with an assessment is infeasible. To address this issue, we may consider introducing new categories of outcomes and/or assessments. For instance we may classify outcomes as examinable although having no guarantee to appear in the examination. In addition we may also design a scheme whereby each category of outcomes has core outcome(s) that is/are definitely assessed (and must be mastered to pass the unit) where as the subordinate outcomes may or may not constitute the assessment.

7 CONCLUSION

We set about the task of designing and implementing outcomes for a Software Engineering unit. In particular, we designed a template whereby we could link each written outcome to where it is assessed, if at all. This allowed us to see which areas were and were not being adequately assessed and consider how the course can be altered appropriately.

In addition, we added a category to each outcome that specified differently levels of achievement. The specification of advanced outcomes will provide more capable students the opportunity to enhance their own abilities and strive for greater knowledge and ability within the various topics taught.

The outcomes can be used as a mechanism for validating the unit content in terms of the curriculum as well at the overall degree programme.

The writing of the outcomes is an ongoing exercise. At this point we continue to improve the relation between the outcomes and the assessment, as well as make alterations to the unit structure as a result of finding gaps in the assessment methods.

REFERENCES

[1] DAZIELL, T. GOURVENEC, S. Partnerships in Outcomes Based Education. In *Proc. 12th Teaching and Learning Forum*. TLF Edith Cowan University, Perth, Australia. 2003. ISBN 0-7298-0548-4.

[2] SPADY, W. Outcomes-based education: Critical issues and answers. American Association of School Administrators, Arlington, V.A. 1994.

[3] KILLEN, R. Outcomes-based Education: Principles: Principles and Possibilities. Unpublished Manuscript. Newcastle, Australia, Faculty of Education, The University of Newcastle, 2000, 1 – 24.

[4] GLATTHORN, A. Outcomes-based education: Reform and the curriculum process. In *Journal of Curriculum and Supervision*. 1993, 354 – 363. ISSN 0882-1232.

[5] *Outcomes-Based Education and Standards For The University of Western Australia* [online]. [Perth, Australia]: Centre for the Advancement of Teaching and Learning, May 2002 [cited 27 February 2004]. Available from World Wide Web: <http://www.catl.uwa.edu.au/obe/OBE-Discussion.doc>.

[6] *A Basic Guide to Writing Student Learning Outcome Statements* [online]. [Perth, Australia]: Centre for the Advancement of Teaching and Learning, 10 September 2003 [cited 4 March 2004]. Available from World Wide Web: <http://www.catl.uwa.edu.au/online/outcomes.html>.

[7] ROMISZOWSKI, A. *The How and Why of Performance Objectives: Preparing Learning Objectives* [online]. [United Kingdom]: World Bank Global Distance Educationet, April 1999 [cited 4 March 2004]. Available from World Wide Web: <http://wbweb4.worldbank.org/DistEd/Teaching/Design/obj-01.html>.

[8] PRIOR, A. *SE104: Software Engineering 104 – Outcomes* [online]. [Perth, Australia] School of Computer Science and Software Engineering, The University of Western Australia, 9 February 2004 [cited 5 March 2004]. Available from World Wide Web: <http://www.csse.uwa.edu.au/~tone/outcomes/outcomes5.html>

[9] PRIOR, A. *SE104: Software Engineering 104 – Outcomes* [online]. [Perth, Australia] School of Computer Science and Software Engineering, The University of Western Australia, 24 January 2004 [cited 5 March 2004]. Available from World Wide Web: <http://www.csse.uwa.edu.au/~tone/outcomes/outcomes12.html>

[10] *Focussing on Outcomes: Outcomes Focus* [online]. [Western Australia], Department of Education and Training, March 2000 [cited 1 March 2004]. Available from World Wide Web: < http://www.eddept.wa.edu.au/outcomes/focus/fc21.htm>